

KWUN BHANSALI LAZARUS LLP
 MICHAEL S. KWUN (SBN 198945)
 mkwun@kblfirm.com
 555 Montgomery St., Suite 750
 San Francisco, CA 94111
 Telephone: 415 630-2350
 Facsimile: 415 367-1539

WUERSCH & GERING LLP
 V. DAVID RIVKIN (admitted *pro hac vice*)
 david.rivkin@wg-law.com
 JOHN A. SMITTEN (admitted *pro hac vice*)
 john.smitten@wg-law.com
 100 Wall St., 10th Fl.
 New York, NY 10005
 Telephone: 212 509-5050
 Facsimile: 212 509-9559

Attorneys for Defendant
 FLORAGUNN GmbH

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
OAKLAND DIVISION

ELASTICSEARCH, INC., a Delaware
 corporation, ELASTICSEARCH B.V., a Dutch
 corporation,

Plaintiffs,

v.

FLORAGUNN GmbH, a German corporation,

Defendant.

CONSOLIDATED CASE
 Case No. 4:19-cv-05553-YGR

ANSWER TO COMPLAINT

Defendant floragunn GmbH (“floragunn”) answers the Complaint of Elasticsearch, Inc.
 and elasticsearch B.V. (together “Elastic”) filed on October 26, 2020, as follows:

Paragraph 1

Allegation: Through its creation and distribution of its Search Guard software,
 floragunn GmbH (“floragunn”) has knowingly and willfully infringed Elasticsearch, Inc.
 and elasticsearch B.V.’s (collectively “Elastic”) copyrights in the source code for
 Elastic’s Elasticsearch X-Pack and Kibana X-Pack software and their predecessors,

Elasticsearch Shield and Kibana Shield. (Unless otherwise specified, Elastic refers to Shield and X-Pack collectively herein as “X-Pack.”)

Response: floragunn denies Elastic’s allegations set forth in paragraph 1 of the Complaint, and further states that it has not infringed any of plaintiff’s source code.

Paragraph 2

Allegation: On September 4, 2019, Elastic brought an action against floragunn to remedy floragunn’s infringement of certain Elastic copyrights in the source code for X-Pack. *See Elasticsearch, Inc. et al. v. floragunn GmbH*, Case No. 4:19-cv-05553-YGR (N.D. Cal.) (“*floragunn I*”). On November 26, 2019, Elastic filed a First Amended Complaint (“FAC”) in the *floragunn I* lawsuit, alleging additional instances of copyright infringement and identifying additional Elastic X-Pack copyrights infringed by floragunn.

Response: floragunn acknowledges that Elastic filed such complaints. floragunn denies that it has infringed any of Elastic’s code.

Paragraph 3

Allegation: In the course of subsequent discovery and investigation, Elastic has identified yet more instances of infringement by floragunn and additional Elastic X-Pack copyrights that floragunn has infringed. Elastic has now registered each of those additional copyrights with the United States Copyright Office.

Response: floragunn acknowledges that Elastic has filed registrations with the United States Copyright Office, but denies that the code in those registrations is eligible for copyright protection. floragunn also denies that it has infringed any of Elastic’s code.

Paragraph 4

Allegation: Elastic files this new lawsuit in light of recent Northern District of California decisions interpreting 17 U.S.C. § 411(a). *See* Order re: Joint Motion for Clarification, ECF No. 59, *UAB “Planner 5D” v. Facebook, Inc.*, No. 19-cv-03132-WHO (N.D. Cal. March 5, 2020); *Izmo, Inc. v. Roadster, Inc.*, No. 18-cv-06092-NC, 2019 WL 2359228 (N.D. Cal. June 4, 2019). Elastic will seek relation of this case to *floragunn I* pursuant to Civil Local Rule 3-12 and consolidation with *floragunn I* for all purposes.

Response: As of the time of filing of this Answer, the two cases have been consolidated.

Paragraph 5

Allegation: Plaintiff Elasticsearch, Inc. is incorporated in Delaware; it has its principal place of business in Mountain View, California. Plaintiff elasticsearch B.V. is incorporated in the Netherlands.

Response: floragunn denies having knowledge or information sufficient to respond to Elastic's allegations set forth in paragraph 5 of the complaint, and therefore denies such allegations.

Paragraph 6

Allegation: Defendant floragunn is a German company with a principal place of business in Berlin, Germany.

Response: floragunn admits the allegations in paragraph 6.

Paragraph 7

Allegation: Elastic's claims for copyright infringement arise under the Copyright Act of 1976, 17 U.S.C. § 101 *et seq.*

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 7 of the complaint.

Paragraph 8

Allegation: This Court has original subject matter jurisdiction of this action under 28 U.S.C. §§ 1331 and 1338.

Response: floragunn admits that this Court has jurisdiction over this action, but denies that it engaged in any infringement of Elastic's code.

Paragraph 9

Allegation: This Court has specific personal jurisdiction over floragunn because, among other reasons, floragunn has extensively offered and distributed its infringing

product containing Elastic's copyrighted material to companies in California and purposefully committed within California the acts from which Elastic's claims arise. Additionally, to the extent floragunn has committed the illegal acts described herein outside of California, it did so knowing and intending that such acts would cause injury to Elastic at its principal place of business within California.

Response: floragunn denies the allegations in paragraph 9, but does not challenge this Court's exercise of personal jurisdiction over floragunn.

Paragraph 10

Allegation: Venue is proper in the Northern District of California under 28 U.S.C. § 1391(b)(2) and 1391(c)(3) because a substantial part of the events or omissions giving rise to the claims alleged in this complaint occurred in this judicial district.

Response: floragunn denies that it engaged in any "events or omissions giving rise to the claims alleged in [the] complaint," but does not challenge the propriety of venue in this district.

Paragraph 11

Allegation: Because this action arises from Elastic's assertion of its intellectual property rights, Northern District of California Civil Local Rule 3-2(c) excludes this action from the division-specific venue rule and subjects this action to assignment on a district-wide basis.

Response: floragunn denies that Elastic's "assertion of its intellectual property rights" has merit, but does not challenge the applicability of Local Rule 3-2(c).

Paragraph 12

Allegation: Elastic produces a core suite of search and analytics products known as the Elastic Stack (formerly known as ELK Stack). The Elastic Stack consists of Elasticsearch, Logstash, Kibana, and Beats. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch. Beats is a family of "data shipper" software that collects and centralizes data that feeds into the other products in Elastic Stack.

Response: floragunn admits the allegations in paragraph 12.

Paragraph 13

Allegation: X-Pack is a set of add-on features to Elastic's core Elastic Stack suite of products. X-Pack includes security, altering, monitoring, reporting, and other add-ons to Elasticsearch, Kibana, and other products in the Elastic Stack. The predecessor to much of X-Pack was known as Shield. (Unless otherwise specified, Elastic refers to Shield and X-Pack collectively herein as "X-Pack.")

Response: floragunn admits the allegations in paragraph 13.

Paragraph 14

Allegation: Elastic has a longstanding commitment to opening the source code underlying many of its products in order to facilitate building a community that helps improve and advance Elastic's products to produce the best software possible. When Elastic releases the source code for its software, it does so under clearly delineated conditions.

Response: (1) floragunn denies the allegations in paragraph in paragraph 14 of the complaint.

Specifically, Elastic misleads when it claims that "it has a longstanding commitment to opening source code underlying many of its products," since it has no "longstanding" commitment to opening source in the case of X-Pack (including Shield) because the code for X-Pack was closed source software from the time it was first released in 2015 until Elastic finally opened the source code to the public in 2018. Before April 2018, it was impossible for third-party developers to contribute anything to the proprietary and closed-source X-Pack code. Search Guard, on the other hand, has been publicly available open code since it was first released in 2015. (2) Second, it is false that "[w]hen Elastic releases the source code for its software, it does so under clearly delineated conditions." For example, Elastic released both its Apache 2 licensed code and its own Elastic-licensed code in the same GitHub repository, causing significant confusion as to which license applied to which files. This practice is commonly referred to as "code mingling" and is highly discouraged by the Open Source community because it leads to situations where a

1 single commit by a developer could contain both Apache2 and Elastic-licensed code. Such
2 commits are called “toxic” for obvious reasons.

3
4 **Paragraph 15**

5 Allegation: In late April 2018, Elastic opened the source code for version 6.2.x of X-
6 Pack. Elastic made the code available on Elastic’s public GitHub code repository for
7 users to inspect, contribute, create issues, and open pull requests, all pursuant to the
8 “Elastic License.” Elastic has released the source code for subsequent versions of X-
9 Pack on GitHub, also under the “Elastic License.”

10 Response: floragunn admits Elastic’s allegation in the first sentence of paragraph 15 of the
11 complaint that Elastic opened the source for version 6.2x of X-Pack in April 2018, but denies
12 that Elastic made the code available solely pursuant to the “Elastic License.” floragunn denies
13 all other allegations in paragraph 15 of the complaint.

14
15 **Paragraph 16**

16 Allegation: The Elastic License did not grant to floragunn or any other party the right to
17 create copies or prepare derivative works for use in any production capacity. And to the
18 extent floragunn acquired any rights pursuant to the Elastic License, those rights
19 terminated immediately and automatically by virtue of floragunn’s actions as described
20 herein. Nor did any license applicable to earlier versions of X-Pack and/or Shield provide
21 floragunn the right to create copies or prepare derivative works for use in any production
22 capacity.

23 Response: floragunn denies the allegations of paragraph 16 of the complaint because the
24 allegation implies that floragunn copied or prepared derivative works of X-Pack or Shield, which
25 it did not. As for the legal interpretation of Elastic’s licenses, floragunn respectfully refers the
26 Court to Elastic’s license to ascertain its terms.

Paragraph 17

Allegation: Elastic is informed and believes, and, on that basis, alleges that floragunn accessed the Elastic code described herein either through decompilation of Elastic binaries, reviewing source-available Elastic repositories, and/or review of otherwise publicly-available Elastic code.

Response: floragunn denies the allegations of paragraph 17.

Paragraph 18

Allegation: floragunn markets and distributes Search Guard, a plug-in for Elasticsearch that offers features similar to the security features that Elastic offers in X-Pack. floragunn makes certain source code for Search Guard available for review and inspection on its GitLab repositories under several different license agreements.

Response: floragunn denies Elastic's allegations in the first sentence of paragraph 18 that floragunn's "*Search Guard, a plug-in for Elasticsearch . . . offers features similar to the security features that Elastic offers in X-Pack.*" In fact, Search Guard always has, and continues to, offer more and different features than X-Pack, although some features are similar.

Paragraph 19

Allegation: Search Guard is available as a "Community Edition" for free for certain uses, but floragunn charges customers for Enterprise and Compliance editions of Search Guard. floragunn prohibits users from, among other things, taking features from the Enterprise or Compliance editions of Search Guard into production without purchasing a license. In fact, floragunn explicitly warned its users that doing so "is illegal" and "can lead to serious legal consequences, which can bring more harm and costs to a company"

Response: floragunn admits the allegations in paragraph 19 of the complaint. Search Guard community edition is available free of charge, and the Enterprise and Compliance editions are available for a fee under difference licenses, which are available to the public for review. floragunn further states that it did not violate the terms of any Elastic license.

Paragraph 20

Allegation: After initiating the *floragunn I* lawsuit, Elastic identified further instances of infringement by floragunn. Infringement by floragunn is evident in at least the

1 following code from a February 13, 2016 commit to the Search Guard
 2 PrivilegesEvaluator.java file:

```

3 private Tuple<Set<String>, Set<String>> resolve(final User user, final String action, final
4   TransportRequest request,
5     final Metadata metaData) {
6
7     if (!(request instanceof CompositeIndicesRequest) && !(request
8       instanceof IndicesRequest)) {
9
10        if (log.isDebugEnabled()) {
11            log.debug("{} is not an IndicesRequest", request.getClass());
12        }
13
14        return new Tuple<Set<String>, Set<String>>(Collections.EMPTY_SET,
15          Collections.EMPTY_SET);
16    }
17
18    final Set<String> indices = new HashSet<String>();
19    final Set<String> types = new HashSet<String>();
20    if (request instanceof CompositeIndicesRequest) {
21        for (final IndicesRequest indicesRequest : ((CompositeIndicesRequest)
22          request).subRequests()) {
23            final Tuple<Set<String>, Set<String>> t = resolve(user, action,
24              indicesRequest, metaData);
25            indices.addAll(t.v1());
26            types.addAll(t.v2());
27        }
28    } else {
29        final Tuple<Set<String>, Set<String>> t = resolve(user, action,
30          (IndicesRequest) request, metaData);
31        indices.addAll(t.v1());
32        types.addAll(t.v2());
33    }
34
35    if (IndexNameExpressionResolver.isAllIndices(new ArrayList<String>(indices)))
36    {
37        indices.clear();
38        indices.add("_all");
39    }
40
41    if (types.isEmpty()) {
42        types.add("_all");
43    }
44
45    return new Tuple<Set<String>,
46      Set<String>>(Collections.unmodifiableSet(indices),
47        Collections.unmodifiableSet(types)); }

```

24 Response: floragunn denies that it has engaged in any copyright infringement and therefore
 25 denies Elastic's allegations in paragraph 20 of the complaint. floragunn further denies that any
 26 code in this commit to PrivilegesEvaluator.java infringes any of Elastic's code.
 27
 28

Paragraph 21

Allegation: The code in Paragraph 20 is copied from and/or is a derivative work of at least the following Elastic code included in the binary of Elasticsearch Shield in the file DefaultIndicesResolver.java, original to Elasticsearch Shield version 1.0.0 Beta1 and released by Elastic on November 3, 2014:

```

public Set<String> resolve(User user, String action, TransportRequest
request, Metadata metaData) {

    boolean isIndicesRequest = request instanceof CompositeIndicesRequest ||
request instanceof IndicesRequest;
    assert isIndicesRequest : "Request [" + request + "] is not an Indices request.
The only requests passing the action matcher should be IndicesRequests";

    // if for some reason we are missing an action... just for safety we'll reject
    if (!isIndicesRequest) {
        return Collections.emptySet();
    }

    if (request instanceof CompositeIndicesRequest) {
        Set<String> indices = Sets.newHashSet();
        CompositeIndicesRequest compositeIndicesRequest =
(CompositeIndicesRequest) request;
        for (IndicesRequest indicesRequest :
compositeIndicesRequest.subRequests()) {
            indices.addAll(resolveIndices(user, action, indicesRequest, metaData));
        }
        return indices;
    }

    return resolveIndices(user, action, (IndicesRequest) request,
metaData); }

```

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 21 of the complaint. floragunn specifically denies that that the commit identified in paragraph 20 is copied from or is a derivative work of any Elastic code. floragunn further denies that any similarities are due to copying of any protectable expression original to Elastic. Moreover, the allegedly infringed code is licensed under Apache Software License 2.0. To the extent that paragraph 21 alleges that floragunn decompiled Elastic object code or otherwise accessed nonpublic code, floragunn denies such allegations.

Paragraph 22

Allegation: Elastic registered Elasticsearch Shield version 1.0.0 Beta1 with the United States Copyright Office on April 30, 2020.

Response: floragunn admits that Elastic submitted such registration, but denies that all the contents of that registration are entitled to copyright protection.

Paragraph 23

Allegation: As shown below, ignoring non-substantive differences in the code and adjusting white space to illustrate similarities, it is clear that at least the floragunn code in Paragraph 20 (on the right) is copied from and/or is a derivative work of at least the Elastic code in Paragraph 21 (on the left):

<pre> public Set<String> resolve(User user, String action, TransportRequest request, MetaData metaData) { boolean isIndicesRequest = request instanceof CompositeIndicesRequest request instanceof IndicesRequest; assert isIndicesRequest : "Request [" + request + "] is not an Indices request. The only requests passing the action matcher should be IndicesRequests"; // If for some reason we are missing an action... just for safety we'll reject if (!isIndicesRequest) { return Collections.emptySet(); } if (request instanceof CompositeIndicesRequest) { Set<String> indices = Sets.newHashSet(); CompositeIndicesRequest compositeIndicesRequest = (CompositeIndicesRequest) request; for (IndicesRequest indicesRequest : compositeIndicesRequest.subRequests()) { indices.addAll(resolveIndices(user, action, indicesRequest, metaData)); } return indices; } return resolveIndices(user, action, (IndicesRequest) request, metaData); } </pre>	<pre> private Tuple<Set<String>, Set<String>> resolve(final User user, final String action, final TransportRequest request, final MetaData metaData) { if (! (request instanceof CompositeIndicesRequest) ! (request instanceof IndicesRequest)) { if (log.isDebugEnabled()) { log.debug("{} is not an IndicesRequest", request.getClass()); } } return new Tuple<Set<String>, Set<String>>(Collections.EMPTY_SET, Collections.EMPTY_SET); final Set<String> indices = new HashSet<String>(); final Set<String> types = new HashSet<String>(); if (request instanceof CompositeIndicesRequest) { for (final IndicesRequest indicesRequest : ((CompositeIndicesRequest) request).subRequests()) { final Tuple<Set<String>, Set<String>> t = resolve(user, action, indicesRequest, metaData); indices.addAll(t.v1()); types.addAll(t.v2()); } } else { final Tuple<Set<String>, Set<String>> t = resolve(user, action, (IndicesRequest) request, metaData); indices.addAll(t.v1()); types.addAll(t.v2()); } if (IndicesWildcardExpressionResolver.isAllIndices(new Array<String>(indices))) { indices.clear(); indices.add("_all"); } if (types.isEmpty()) { types.add("_all"); } return new Tuple<Set<String>, Set<String>>(Collections.unmodifiableSet(indices), Collections.unmodifiableSet(types)); } </pre>
---	---

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 23 of the complaint. floragunn specifically denies that that the commit identified in paragraph 20 is copied from or is a derivative work of any Elastic code, much less that it is "clear that . . . [it] is copied from and/or is a derivative work of at least the Elastic code in Paragraph 21." floragunn further denies that any similarities are due to copying of any protectable expression original to Elastic. To the extent that paragraph 23 alleges that floragunn decompiled Elastic object code or otherwise accessed nonpublic code, floragunn denies such allegations.

Paragraph 24

Allegation: Infringement by floragunn is also evident in an April 5, 2017 commit to the Search Guard Kibana plugin login_controller.js file. That file contains the following code:

```
const {query, hash} = parse($window.location.href, true);
let nextUrl;
if (query.next) {
  nextUrl = query.next + (hash || "")
} else {
  nextUrl = "/";
}
```

Response: floragunn admits that the April 5, 2017 commit contained such code. floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's remaining allegations in paragraph 24 of the complaint. floragunn further denies that any code in this commit to login_controller.js infringes any of Elastic's code.

Paragraph 25

Allegation: That floragunn code is copied from and/or is a derivative work of the following Elastic code included in parse_next.js and original to Kibana Shield version 2.3.2 released by Elastic on April 26, 2016, reproduced here:

```
const {query, hash} = parse(location.href, true);  
if (query.next) return query.next + (hash || "");  
return '/';
```

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 25 of the complaint. floragunn specifically denies that the commit identified in paragraph 24 is copied from or is a derivative work of any Elastic code. floragunn further denies that any similarities are due to copying of any protectable expression original to Elastic. Moreover, the allegedly infringed code is licensed under the Apache Software License 2.0.

Paragraph 26

Allegation: Elastic registered Kibana Shield version 2.3.2 with the United States Copyright Office on November 6, 2019.

Response: floragunn admits that Elastic submitted such a registration, but denies that all of the contents of the registration are entitled to copyright protection.

Paragraph 27

Allegation: Infringement by floragunn is also evident in an August 6, 2017 commit to the Search Guard Kibana plugin index.html and filterbar.html files. Those files contain the following code:

filterbar.html Lines 2-32	<pre> <div class="kuiToolBar"> <div class="kuiToolBarSearch"> <div class="kuiToolBarSearchBox"> <div class="kuiToolBarSearchBox__icon kuiIcon fa-search"></div> <input class="kuiToolBarSearchBox__input ng-pristine ng-untouched ng-valid" type="text" placeholder="Search..." ng-model="query"> </div> </div> <div class="kuiToolBarSection"> <a ng-click="new()" class="kuiButton kuiButton--primary kuiButton--iconText"> Add <a ng-href="#/" class="kuiButton kuiButton--basic kuiButton--iconText"> Back </div> <div class="kuiToolBarSection"> <!-- We need an empty section for the buttons to be positioned consistently. --> </div> </div> <!-- NoResults --> <div class="kuiPanel kuiPanel--centered ng-hide" ng-show="!(resourcenames filter:query).length"> <div class="kuiNoItems"> No matching entries found. </div> </div> </pre>
index.html Line 13	<pre> <table class="kuiTable" ng-show="(resourcenames filter:query).length"> </pre>
index.html Line 34	<pre> <tr ng-repeat="username in resourcenames filter:query" data-test-subj="userRow" class="kuiTableRow"> </pre>
index.html Lines 44-48	<pre> <td class="kuiTableRowCell" data-test-subj="userRowRoles"> <div class="kuiTableRowCell__liner"> <div ng-repeat="role in resources[username].roles"> { {role} }
</div> </div> </td> </pre>

Response: floragunn admits that the August 6, 2017 commit contains such code. floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 27 of the complaint. floragunn further denies that any code in this commit to index.html and filterbar.html infringes any of Elastic's code.

Paragraph 28

Allegation: That floragunn code is copied from and/or is a derivative work of Elastic code in the users.html file original to Kibana X-Pack versions 5.0.0 and 5.2.0 and released by Elastic on or before January 31, 2017, reproduced here:

Lines 22–70	<pre> <div class="kuiToolBar"> <div class="kuiToolBarSearch"> <div class="kuiToolBarSearchBox"> <div class="kuiToolBarSearchBox__icon kuiIcon fa- search"></div> <input class="kuiToolBarSearchBox__input" type="text" placeholder="Search..." aria-label="Filter" ng-model="query" > </div> </div> <div class="kuiToolBarSection"> <!-- Delete users button --> <button ng-click="deleteUsers()" class="kuiButton kuiButton--danger kuiButton--iconText" ng-if="selectedUsers.length" > Delete </button> <!-- Create user button --> <a href="#/management/elasticsearch/users/edit" ng-click="newUser()" class="kuiButton kuiButton--primary kuiButton--iconText" ng-if="!selectedUsers.length" data-test-subj="createUserButton" </pre>
-------------	--

1	>
2	
3	Create user
4	
5	</div>
6	<div class="kuiToolBarSection">
7	<!-- We need an empty section for the buttons to be positioned consistently. -->
8	</div>
9	</div>
10	<!-- NoResults -->
11	<div class="kuiPanel kuiPanel--centered" ng-show="!(users filter:query).length">
12	<div class="kuiNoItems">
13	No matching users found.
14	</div>
15	</div>
16	Line 73 <table class="kuiTable" ng-show="(users filter:query).length">
17	Lines 135-39 <tr
18	ng-repeat="user in users orderBy:'username' filter:query
19	orderBy:sort.orderBy:sort.reverse"
20	data-test-subj="userRow"
21	class="kuiTableRow"
22	>
23	Lines 176-82 <td class="kuiTableRowCell" data-test-subj="userRowRoles">
24	<div class="kuiTableRowCell_liner">
25	
26	<a class="kuiLink" ng-
27	href="#/management/elasticsearch/roles/edit/{{role}}">{{ role
28	}},
	
	</div>
	</td>

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 28 of the complaint. floragunn specifically denies that the commit identified in paragraph 27 is copied from or is a derivative work of any Elastic code. floragunn further denies that any similarities are due to copying of any protectable expression original to Elastic. Moreover, the allegedly infringed code is licensed under the Apache Software License 2.0.

Paragraph 29

Allegation: As shown below, ignoring non-substantive differences in the code and adjusting white space to illustrate similarities, it is clear that at least the floragunn code in Paragraph 27 is copied from and/or is a derivative work of at least the Elastic code in Paragraph 28. This comparison includes additional indicia of copying in the following non-functional comment: "<!-- We need an empty section for the buttons to be positioned consistently. -->"

(a) filterbar.html lines 2–32 (on the right) and users.html lines 22–70 (on the left):

```

<div class="kuiToolbar">
  <div class="kuiToolbarSearch">
    <div class="kuiToolbarSearchBox">
      <div class="kuiToolbarSearchBox__icon kuiIcon fa-search"></div>
      <input
        class="kuiToolbarSearchBox__input"
        type="text"
        placeholder="Search..."
        aria-label="Filter"
        ng-model="query"
      />
    </div>
  </div>

  <div class="kuiToolbarSection">
    <!-- Delete users button -->
    <button
      ng-click="deleteUsers()"
      class="kuiButton kuiButton--danger kuiButton--iconText"
      ng-if="selectedUsers.length"
    >
      <span class="kuiButton__icon kuiIcon fa-trash"></span>
      Delete
    </button>
    <!-- Create user button -->
    <a
      href="#/management/elasticsearch/users/edit"
      ng-click="addUser()"
      class="kuiButton kuiButton--primary kuiButton--iconText"
      ng-if="!selectedUsers.length"
      data-test-subj="createUserButton"
    >
      <span class="kuiButton__icon kuiIcon fa-plus"></span>
      Create user
    </a>
  </div>

  <div class="kuiToolbarSection">
    <!-- We need an empty section for the buttons to be positioned correctly -->
    </div>
  </div>
</div>

<div class="kuiPanel kuiPanel--centered" ng-show="(!users | filter:query).length">
  <div class="kuiNoItems">
    No <span ng-show="query" <!-- matching --></span> users found.
  </div>
</div>

```

```

<div class="kuiToolbar">
  <div class="kuiToolbarSearch">
    <div class="kuiToolbarSearchBox">
      <div class="kuiToolbarSearchBox__icon kuiIcon fa-search"></div>
      <input
        class="kuiToolbarSearchBox__input"
        ng-pristine
        ng-untouched
        ng-val="0"
        type="text"
        placeholder="Search..."
        ng-model="query"
      />
    </div>
  </div>

  <div class="kuiToolbarSection">
    <a
      ng-click="add()"
      class="kuiButton kuiButton--primary kuiButton--iconText"
    >
      <span class="kuiButton__icon kuiIcon fa-plus"></span>
      Add
    </a>
    <a
      href="#/"
      class="kuiButton kuiButton--basic kuiButton--iconText"
    >
      <span class="kuiButton__icon kuiIcon fa-chevron-left"></span>
      Back
    </a>
  </div>

  <div class="kuiToolbarSection">
    <!-- We need an empty section for the buttons to be positioned correctly -->
    </div>
  </div>
</div>

<div class="kuiPanel kuiPanel--centered ng-hide" ng-show="(!resourceNames | filter:query).length">
  <div class="kuiNoItems">
    No <span ng-show="query" <!-- matching --></span> entries found.
  </div>
</div>

```

(b) index.html line 13 (on the right) and users.html line 73 (on the left):

```

<table class="kuiTable" ng-show="(!users | filter:query).length">
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th>Role</th>
    <th>Actions</th>
  </tr>
  <tr>
    <td>{{user.username}}</td>
    <td>{{user.email}}</td>
    <td>{{user.role}}</td>
    <td>
      <button class="kuiButton kuiButton--iconText" ng-click="deleteUser(user)">
        <span class="kuiButton__icon kuiIcon fa-trash"></span>
      </button>
    </td>
  </tr>
</table>

```

```

<table class="kuiTable" ng-show="(!resourceNames | filter:query).length">
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th>Role</th>
    <th>Actions</th>
  </tr>
  <tr>
    <td>{{resourceName}}</td>
    <td>{{resourceEmail}}</td>
    <td>{{resourceRole}}</td>
    <td>
      <button class="kuiButton kuiButton--iconText" ng-click="deleteResource(resourceName)">
        <span class="kuiButton__icon kuiIcon fa-trash"></span>
      </button>
    </td>
  </tr>
</table>

```

(c) index.html Line 34 (on the right) and users.html Lines 135–139 (on the left):

```

<tr>
  <th>Name</th>
  <th>Email</th>
  <th>Role</th>
  <th>Actions</th>
</tr>
<tr>
  <td>{{user.username}}</td>
  <td>{{user.email}}</td>
  <td>{{user.role}}</td>
  <td>
    <button class="kuiButton kuiButton--iconText" ng-click="deleteUser(user)">
      <span class="kuiButton__icon kuiIcon fa-trash"></span>
    </button>
  </td>
</tr>

```

```

<tr>
  <th>Name</th>
  <th>Email</th>
  <th>Role</th>
  <th>Actions</th>
</tr>
<tr>
  <td>{{resourceName}}</td>
  <td>{{resourceEmail}}</td>
  <td>{{resourceRole}}</td>
  <td>
    <button class="kuiButton kuiButton--iconText" ng-click="deleteResource(resourceName)">
      <span class="kuiButton__icon kuiIcon fa-trash"></span>
    </button>
  </td>
</tr>

```


(d) index.html Lines 44–48 (on the right) and users.html Lines 176–182 (on the left):

<pre> <td class="kuiTableRowCell" data-test-subj="userRowRoles"> <div class="kuiTableRowCell_line"> {{ role }}, </div> </td> </pre>	<pre> <td class="kuiTableRowCell" data-test-subj="userRowRoles"> <div class="kuiTableRowCell_line"> <div ng-repeat="role in resources[username].roles"> {{role}}
</div> </div> </td> </pre>
--	---

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic’s allegations in paragraph 29 of the complaint. floragunn specifically denies that the commit identified in paragraph 27 is copied from or is a derivative work of any Elastic code, much less that it is “clear that . . . [it] is copied from and/or is a derivative work of at least the Elastic code in Paragraph 28.” floragunn further denies that any similarities are due to copying of any protectable expression original to Elastic.

Paragraph 30

Allegation: Elastic registered Kibana X-Pack versions 5.0.0 and 5.2.0 with the United States Copyright Office on April 30, 2020 and September 19, 2019, respectively.

Response: floragunn admits that Elastic made these registrations, but denies that they entitle Elastic to copyright protection for the code that Elastic claims to be infringed.

Paragraph 31

Allegation: Further newly identified infringement by floragunn is evident in a September 11, 2017 commit to the Search Guard PrivilegesEvaluator.java file, including at least:

```
if(action.equals("indices:data/write/bulk[s]")) {
```

and the following commented-out—that is, non-functional—code:

```

if(request instanceof BulkRequest) {
    for(DocWriteRequest<?> ar: ((BulkRequest) request).requests()) {
        //require also op type permissions
        switch(ar.opType()) {

```

```

1      case CREATE: additionalPermissionsRequired.add(IndexAction.NAME);break;
2      case INDEX: additionalPermissionsRequired.add(IndexAction.NAME);break;
3      case DELETE:
4      additionalPermissionsRequired.add(DeleteAction.NAME);break; case
5      UPDATE: additionalPermissionsRequired.add(UpdateAction.NAME);break; }
6
7      }
8
9      }

```

Response: floragunn admits that the commit contained such changes, but denies that it infringed any of Elastic's copyrights. floragunn specifically denies that any similarities to Elastic's code are due to copying of any protectable expression original to Elastic.

Paragraph 32

Allegation: A subsequent September 26, 2017 commit to the Search Guard PrivilegesEvaluator.java file contains a further example of floragunn's infringement. In that commit, floragunn retained the infringing commented-out code quoted in Paragraph 31, but removed the “/*” and “*/” symbols that had commented out the infringing code.

Response: floragunn admits that the commit contained such changes, but denies that it infringed any of Elastic's copyrights. floragunn specifically denies that any similarities to Elastic's code are due to copying of any protectable expression original to Elastic.

Paragraph 33

Allegation: Then, floragunn again infringed Elastic's copyrights in an October 1, 2017 commit to the Search Guard PrivilegesEvaluator.java file. That commit added the following code:

```

21      switch (bir.request().opType()) {
22      case CREATE:
23      additionalPermissionsRequired.add(IndexAction.NAME);
24      break;
25      case INDEX:
26      additionalPermissionsRequired.add(IndexAction.NAME);
27      break;
28      case DELETE:
29      additionalPermissionsRequired.add(DeleteAction.NAME);
30      break;
31      case UPDATE:
32      additionalPermissionsRequired.add(UpdateAction.NAME);
33      break;

```

The same floragunn commit and file also commented out the following

```
code: /*if(request instanceof BulkRequest) {

for(DocWriteRequest<?> ar: ((BulkRequest) request).requests()) {
//require also op type permissions
switch(ar.opType()) {
case CREATE: additionalPermissionsRequired.add(IndexAction.NAME);break;
case INDEX: additionalPermissionsRequired.add(IndexAction.NAME);break;
case DELETE:
additionalPermissionsRequired.add(DeleteAction.NAME);break; case
UPDATE: additionalPermissionsRequired.add(UpdateAction.NAME);break; }

}
}*/
```

Response: floragunn admits that the commit contained such code, but denies that it infringes any of Elastic's copyrights. floragunn specifically denies that any similarities are due to copying of any protectable expression original to Elastic.

Paragraph 34

Allegation: An October 10 commit to the same file demonstrates further infringement by floragunn. That commit removed the commented-out infringing code in Paragraph 33 but retained other infringing code. Additionally, floragunn's October 10 commit contains code that is a derivative work of Elastic's copyrighted source code.

Response: floragunn admits that the October 10 commit contained that change.

floragunn denies that the commit infringed any of Elastic's copyrights. floragunn specifically denies that any similarities are due to copying of any protectable expression original to Elastic. floragunn further denies that any code in the commit identified in this paragraph constitutes a derivative work of Elastic's code.

Paragraph 35

Allegation: The floragunn commits referenced in Paragraphs 31–34 contain code that is copied from and/or is a derivative work of at least the following Elastic code included in X-Pack in the file AuthorizationService.java, original to Elasticsearch X-Pack 5.6.0 and released by Elastic on September 11, 2017:

```
if (action.equals(TransportShardBulkAction.ACTION_NAME)) {
```

And:

```

1      final DocWriteRequest docWriteRequest = item.request();
2      switch (docWriteRequest.opType()) {
3          case INDEX:
4          case CREATE:
5              return IndexAction.NAME;
6          case UPDATE:
7              return UpdateAction.NAME;
8          case DELETE:
9              return DeleteAction.NAME;
10         }

```

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 35 of the complaint. floragunn specifically denies that that the commits identified in paragraphs 31-34 are copied from or are a derivative work of any Elastic code. floragunn further denies that any similarities are due to copying of any protectable expression original to Elastic.

Paragraph 36

Allegation: Elastic had not publicly released the source code for X-Pack alleged in Paragraph 35 at the time of floragunn's copying and/or creation of derivative works from that code. Elastic is informed and believes and, on that basis, alleges that floragunn decompiled Elastic's binaries or otherwise gained access to Elastic's source code to create the copies and/or derivative works referenced in Paragraphs 31–34. Further, Elastic is informed and believes and, on that basis, alleges that the decompilation process employed by floragunn caused the code originally written by Elastic as:

```

19      if (action.equals(TransportShardBulkAction.ACTION_NAME)) {

```

to be rendered as:

```

21      if(action.equals("indices:data/write/bulk[s]")) {

```

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 36 of the complaint. floragunn further denies that it decompiled Elastic's binaries or otherwise accessed any non-public Elastic code. floragunn specifically denies that that the commits identified in paragraphs 31-34 are copied from or are a derivative work of any Elastic code. floragunn further denies that any similarities are due to copying of any protectable expression original to Elastic.

Paragraph 37

Allegation: Elastic is informed and believes and, on that basis, alleges that the creation and inclusion of the decompiled code referenced in Paragraphs 31 and 36 is both an act of infringement and evidence of infringing acts by floragunn.

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 37 of the complaint. floragunn further denies that it decompiled Elastic's binaries or otherwise accessed any non-public Elastic code.

Paragraph 38

Allegation: Elastic registered Elasticsearch X-Pack 5.6.0 with the United States Copyright Office on October 13, 2020.

Response: floragunn admits that Elastic made such a registration, but denies that all its contents are entitled to copyright protection.

Paragraph 39

Allegation: After initiating the *floragunn I* lawsuit, Elastic identified further infringement by floragunn in a June 7, 2018 commit to Search Guard in the file DlsFlsValveImpl.java. Infringement by floragunn is evident in at least the following code from the Search Guard DlsFlsValveImpl.java file:

```
if(request instanceof BulkShardRequest) {
    for(BulkItemRequest inner:((BulkShardRequest) request).items()) {
        if(inner.request() instanceof UpdateRequest) {
            listener.onFailure(new ElasticsearchSecurityException("Update is not
supported when FLS or DLS is activated"));
            return false;
        }
    }
}
```

Response: floragunn admits that such code appears in the commit, but denies that it infringes any of Elastic's code. floragunn specifically denies that any similarity to Elastic's code is the result of copying by floragunn.

Paragraph 40

Allegation: The floragunn code in Paragraph 39 is copied from and/or is a derivative work of at least the following Elastic code included in the file BulkShardRequestInterceptor.java, original to Elasticsearch X-Pack version 5.1.1 and released on December 8, 2016:

```

        for (BulkItemRequest bulkItemRequest : request.items()) {
            IndicesAccessControl.IndexAccessControl indexAccessControl =
indicesAccessControl.getIndexPermissions(bulkItemRequest.index());
            if (indexAccessControl != null) {
                boolean fls =
indexAccessControl.getFieldPermissions().hasFieldLevelSecurity();
                boolean dls = indexAccessControl.getQueries() != null;
                if (fls || dls) {
                    if (bulkItemRequest.request() instanceof UpdateRequest) {
                        throw new ElasticsearchSecurityException("Can't execute a bulk request
with update requests embedded if " +
                            "field or document level security is enabled",
RestStatus.BAD_REQUEST);
                    }
                }
            }
            logger.trace("intercepted bulk request for index [{}] without any update requests,
continuing execution",
                bulkItemRequest.index());
        }
    }

    @Override
    public boolean supports(TransportRequest request) {
        return request instanceof BulkShardRequest;
    }
}

```

Response: floragunn denies that it copied or created derivative works, and on that basis denies the first sentence of this allegation. floragunn specifically denies that any similarity between its June 7, 2018 commit and Elastic's code is a result of copying.

Paragraph 41

Allegation: Elastic registered Elasticsearch X-Pack version 5.1.1 with the United States Copyright Office on October 13, 2020.

Response: floragunn admits that Elastic made such a registration, but denies that all its contents are entitled to copyright protection.

Paragraph 42

Allegation: Further infringement by floragunn is evident in at least the following commented-out code from the June 7, 2018 commit to the Search Guard DlsFlsValveImpl.java file:

```
aliasRequest.getAliasActions().stream().filter(a->a.actionType()
== Type.ADD).forEach(a->{
```

And:

```
listener.onFailure(new ElasticsearchSecurityException("Managing aliases is not
supported when FLS or DLS is activated"));
```

Response: floragunn denies that it infringed any of Elastic's code, much less that any infringement is "evident." floragunn admits that this code appears in the June 7, 2018 commit.

Paragraph 43

Allegation: The floragunn code in Paragraph 42 is copied from and/or is a derivative work of at least the following Elastic code included the file IndicesAliasesRequestInterceptor.java, released December 19, 2017 and original to Elasticsearch X-Pack version 6.1.1:

```
for (IndicesAliasesRequest.AliasActions aliasAction : request.getAliasActions())
{ if (aliasAction.actionType() ==
IndicesAliasesRequest.AliasActions.Type.ADD) {
```

And:

```
throw new ElasticsearchSecurityException("Alias requests are not allowed for users
who have " +
"field or document level security enabled on one of the indices",
RestStatus.BAD_REQUEST);
```

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works and therefore denies Elastic's allegations in paragraph 43 of the complaint.

Paragraph 44

Allegation: Elastic registered Elasticsearch X-Pack version 6.1.1 with the United States Copyright Office on October 13, 2020.

Response: floragunn admits that Elastic submitted such a registration, but denies that all its contents are entitled to copyright protection.

Paragraph 45

Allegation: Further infringement by floragunn is evident in a June 4, 2019 commit to the Search Guard searchguard_saved_objects_client.js file. That file contains the following code:

```
const ACTIONS = {
  CREATE: 'create',
  BULK_CREATE: 'bulk_create',
  FIND: 'find',
  GET: 'get',
  BULK_GET: 'bulk_get', // @todo Why the snake case here? What do our permissions
  look like
  UPDATE: 'update',
  'DELETE': 'delete',
};
export default class SearchguardSavedObjectsClient {
  constructor(request, searchguardBackend, requestRepository, savedObjects) {
    this.request = request;
    this.searchguardBackend = searchguardBackend;
    this.callWithRequestRepository = requestRepository;
    this.errors = savedObjects.SavedObjectsClient.errors;
    this.savedObjectTypes = savedObjects.types;
  }
  async create(type, attributes = {}, options = {}) {
    return await this.authorize(
      type,
      ACTIONS.CREATE,
      repository => repository.create(type, attributes, options)
    );
  }
  async bulkCreate(objects, options = {}) {
    let types = [];
    objects.forEach((object) => {

      if (types.indexOf(object.type) === -1) {
        types.push(object.type);
      }
    });

    return await this.authorize(
      types,
      ACTIONS.BULK_CREATE,
      repository => repository.bulkCreate(objects, options)
    );
  }
}
```



```

1  async find(options = {}) {
2      return await this.authorize(
3          options.type,
4          ACTIONS.FIND,
5          repository => repository.find(options)
6      );
7  }
8  async get(type, id) {
9      return await this.authorize(
10         type,
11         ACTIONS.GET,
12         repository => repository.get(type, id)
13     );
14 }
15 async bulkGet(objects = []) {
16     let types = [];
17     objects.forEach((object) => {
18         if (types.indexOf(object.type) === -1) {
19             types.push(object.type);
20         }
21     });
22     return await this.authorize(
23         types,
24         ACTIONS.BULK_GET,
25         repository => repository.bulkGet(objects)
26     );
27 }
28 async update(type, id, attributes, options = {}) {
29     return await this.authorize(
30         type,
31         ACTIONS.UPDATE,
32         repository => repository.update(type, id, attributes, options)
33     );
34 }
35 async delete(type, id) {
36     return await this.authorize(
37         type,
38         ACTIONS['DELETE'],
39         repository => repository.delete(type, id)
40     );
41 }
42 buildPermissionName(action, type) {
43     return `kibana:saved_objects/${type}/${action}`;
44 }
45
46 getPermissionInfo(type, action) {
47     const types = Array.isArray(type) ? type : [type];
48     let permissionToType = {};
49     const permissionNames = types.map((type) => {
50         const permissionName = this.buildPermissionName(action, type);
51         permissionToType[permissionName] = type;
52         return permissionName;
53     });
54 }

```

```

1      return {
2          types,
3          permissionNames,
4          permissionToType,
5          permissionsParameter: permissionNames.join(',')
6      };
7  }
8  async authorize(type, action, clientCallback) {
9      const permissionsInfo = this.getPermissionInfo(type, action);
10     const permissionsResult = await
11     this.checkPermissions(permissionsInfo.permissionsParameter,
12     permissionsInfo.permissionToType);
13
14     if (permissionsResult.hasAllPermissions !== true) {
15         const errorMessage = `Unauthorized: ${action} for ${permissionsInfo.types.join(',
16         ')}, missing permissions: ${permissionsResult.missingPermissions.join(', ')}';
17         // @todo Error handling in the frontend
18         throw this.errors.decorateForbiddenError(new Error(errorMessage));
19     } else {
20         return await clientCallback(this.callWithRequestRepository);
21     }
22 }
23
24     async checkPermissions(permissionsParameter, permissionToType) {
25         try {
26             const backendResult = await
27             this.searchguardBackend.hasPermissions(this.request.headers, permissionsParameter);
28
29             let checkResult = {
30                 hasAllPermissions: false,
31                 missingPermissions: [],
32                 missingTypes: [],
33                 allowedTypes: [],
34             };
35
36             // Go through the response for each permission
37             for (let permission in backendResult.permissions) {
38                 let permissionType = permissionToType[permission];
39                 if (backendResult.permissions[permission] !== true) {
40                     checkResult.missingPermissions.push(permission);
41                     checkResult.missingTypes.push(permissionType);
42                 } else if (backendResult.permissions[permission] === true) {
43                     checkResult.allowedTypes.push(permissionType)
44                 }
45             }
46
47             if (checkResult.missingPermissions.length) {
48
49                 checkResult.hasAllPermissions = false;
50                 return checkResult;
51             } else {
52                 checkResult.hasAllPermissions = true;
53                 return checkResult;
54             }
55         }
56     }

```

```

1      } catch (error) {
2          throw this.errors.decorateGeneralError(error, 'Could not check for application
3      permissions');
4      }
5  }
6  }
7  }

```

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works, and therefore denies the allegations in paragraph 45 of the complaint.

floragunn admits that the commit identified in this paragraph contains such code.

Paragraph 46

Allegation: As further alleged below, at least substantial portions of that floragunn code is copied from and/or is a derivative work of following Elastic code released in a July 24, 2018 commit to `secure_saved_objects_client.js` and original to Kibana X-Pack 6.4.0, reproduced here:

```

14 export class SecureSavedObjectsClient {
15     constructor(options) {
16         const {
17             errors,
18             internalRepository,
19             callWithRequestRepository,
20             checkPrivileges,
21             auditLogger,
22             savedObjectTypes,
23             actions,
24         } = options;
25         this.errors = errors;
26         this._internalRepository = internalRepository;
27         this._callWithRequestRepository = callWithRequestRepository;
28         this._checkPrivileges = checkPrivileges;
29         this._auditLogger = auditLogger;
30         this._savedObjectTypes = savedObjectTypes;
31         this._actions = actions;
32     }
33     async create(type, attributes = {}, options = {}) {
34         return await this._execute(
35             type,
36             'create',
37             { type, attributes, options },
38             repository => repository.create(type, attributes, options),
39         );
40     }
41     async bulkCreate(objects, options = {}) {
42         const types = uniq(objects.map(o => o.type));
43         return await this._execute(

```

```

1      types,
2      'bulk_create',
3      { objects, options },
4      repository => repository.bulkCreate(objects, options),
5
6      );
7      }
8      async delete(type, id) {
9      return await this._execute(
10         type,
11         'delete',
12         { type, id },
13         repository => repository.delete(type, id),
14     );
15     }
16     async find(options = {}) {
17     if (options.type) {
18         return await this._findWithTypes(options);
19     }
20
21     return await this._findAcrossAllTypes(options);
22     }
23     async bulkGet(objects = []) {
24     const types = uniq(objects.map(o => o.type));
25     return await this._execute(
26         types,
27         'bulk_get',
28         { objects },
29         repository => repository.bulkGet(objects)
30     );
31     }
32     async get(type, id) {
33     return await this._execute(
34         type,
35         'get',
36         { type, id },
37         repository => repository.get(type, id)
38     );
39     }
40     async update(type, id, attributes, options = {}) {
41     return await this._execute(
42         type,
43         'update',
44         { type, id, attributes, options },
45         repository => repository.update(type, id, attributes, options)
46     );
47     }
48     async _checkSavedObjectPrivileges(actions) {
49     try {
50         return await this._checkPrivileges(actions);
51     } catch(error) {
52         const { reason } = get(error, 'body.error', {});
53         throw this.errors.decorateGeneralError(error, reason);

```

```

1      }
2      }
3      async _execute(typeOrTypes, action, args, fn) {
4      const types = Array.isArray(typeOrTypes) ? typeOrTypes : [typeOrTypes];
5      const actions = types.map(type => this._actions.getSavedObjectAction(type, action));
6      const { result, username, missing } = await this._checkSavedObjectPrivileges(actions);
7      switch (result) {
8
9          case CHECK_PRIVILEGES_RESULT.AUTHORIZED:
10             this._auditLogger.savedObjectsAuthorizationSuccess(username, action, types, args);
11             return await fn(this._internalRepository);
12          case CHECK_PRIVILEGES_RESULT.LEGACY:
13             return await fn(this._callWithRequestRepository);
14          case CHECK_PRIVILEGES_RESULT.UNAUTHORIZED:
15             this._auditLogger.savedObjectsAuthorizationFailure(username, action, types,
16             missing, args);
17             const msg = `Unable to ${action} ${[...types].sort().join(',')}, missing
18             ${[...missing].sort().join(',')}`;
19             throw this.errors.decorateForbiddenError(new Error(msg));
20             default:
21                 throw new Error('Unexpected result from hasPrivileges');
22             }
23         }
24     }
25     async _findAcrossAllTypes(options) {
26         const action = 'find';
27         // we have to filter for only their authorized types
28         const types = this._savedObjectTypes;
29         const typesToPrivilegesMap = new Map(types.map(type => [type,
30         this._actions.getSavedObjectAction(type, action)]));
31         const { result, username, missing } = await
32         this._checkSavedObjectPrivileges(Array.from(typesToPrivilegesMap.values()));
33         if (result === CHECK_PRIVILEGES_RESULT.LEGACY) {
34             return await this._callWithRequestRepository.find(options);
35         }
36         const authorizedTypes = Array.from(typesToPrivilegesMap.entries())
37         .filter(([ , privilege]) => !missing.includes(privilege))
38         .map(([type]) => type);
39         if (authorizedTypes.length === 0) {
40             this._auditLogger.savedObjectsAuthorizationFailure(
41                 username,
42                 action,
43                 types,
44                 missing,
45                 { options }
46             );
47             throw this.errors.decorateForbiddenError(new Error(`Not authorized to find
48             saved_object`));
49         }
50         this._auditLogger.savedObjectsAuthorizationSuccess(username, action,
51         authorizedTypes, { options });
52         return await this._internalRepository.find({
53             ...options,
54             type: authorizedTypes

```

```

    });
  }
  async _findWithTypes(options) {
    return await this._execute(
      options.type,
      'find',
      { options },
      repository => repository.find(options)
    );
  }
}

```

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works and therefore denies Elastic's allegations in paragraph 46. floragunn specifically denies that any similarities are due to copying of any protectable expression original to Elastic.

Paragraph 47

Allegation: As shown below, adjusting the order of the code in Paragraphs 45 and 46 to illustrate similarities, it is clear that at least the floragunn code shown below (on the right) is copied from and/or is a derivative work of at least the Elastic code shown below (on the left):

<pre> export class SecureSavedObjectsClient { constructor(options) { const { errors, internalRepository, callWithRequestRepository, checkPrivileges, auditLogger, savedObjectTypes, actions, } = options; </pre>	<pre> export default class SearchguardSavedObjectsClient { constructor(request, searchguardBackend, requestRepository, savedObjects) { this.request = request; this.searchguardBackend = searchguardBackend; this.callWithRequestRepository = requestRepository; this.errors = savedObjects.SavedObjectsClient.errors; this.savedObjectTypes = savedObjects.types; } </pre>
<pre> async create(type, attributes = {}, options = {}) { return await this._execute(type, 'create', { type, attributes, options }, repository => </pre>	<pre> async create(type, attributes = {}, = {}) { return await this.authorize(type, ACTIONS.CREATE, repository => </pre>

```

1 repository.create(type, attributes,
2 options),
3 );
4 }

```

```

1 repository.create(type, attributes,
2 options)
3 );
4 }

```

```

5 async bulkCreate(objects, options = {})
6 {
7     const types = uniq(objects.map(o =>
8 o.type));
9
10     return await this._execute(
11         types,

```

```

5 async bulkCreate(objects, options = {}),
6     let types = [];
7     objects.forEach((object) => {
8         if (types.indexOf(object.type)
9 === -1) {
10             types.push(object.type);
11         }
12     });
13
14     return await this.authorize(
15         types,

```

```

12 'bulk_create',
13 { objects, options },
14 repository =>
15 repository.bulkCreate(objects,
16 options),
17 );
18 }

```

```

12 ACTIONS.BULK_CREATE,
13
14 repository =>
15 repository.bulkCreate(objects, options)
16 );
17 }

```

```

17 async delete(type, id) {
18     return await this._execute(
19         type,
20         'delete',
21         { type, id },
22         repository =>
23 repository.delete(type, id),
24 );
25 }

```

```

17 async delete(type, id) {
18     return await this.authorize(
19         type,
20         ACTIONS['DELETE'],
21         repository =>
22 repository.delete(type, id)
23 );
24 }

```

```

1  async find(options = {}) {
2      if (options.type) {
3          return await
4          this._findWithTypes(options);
5      }
6      return await
7      this._findAcrossAllTypes(options);
8  }

```

```

1  async find(options = {}) {
2      return await this.authorize(
3          options.type,
4          ACTIONS.FIND,
5          repository =>
6          repository.find(options)
7      );
8  }

```

```

9  async bulkGet(objects = []) {
10     const types = uniq(objects.map(o =>
11     o.type));
12
13     return await this._execute(
14         types,
15         'bulk_get',
16         { objects },
17         repository =>
18         repository.bulkGet(objects)
19     );
20 }

```

```

9  async bulkGet(objects = []) {
10     let types =
11         [
12         ];
13     objects.forEach((object) => {
14         if (types.indexOf(object.type)
15         === -1) {
16             types.push(object.type);
17         }
18     });
19     return await this.authorize(
20         types,
21         ACTIONS.BULK_GET,
22         repository =>
23         repository.bulkGet(objects)
24     );
25 }

```

```

26 async get(type, id) {
27     return await this._execute(
28         type,
29         'get',
30         { type, id },
31         repository => repository.get(type,
32         id)
33     );
34 }

```

```

26 async get(type, id) {
27     return await this.authorize(
28         type,
29         ACTIONS.GET,
30         repository => repository.get(type,
31         id)
32     );
33 }

```

```

35 async update(type, id, attributes,
36 options = {}) {
37     return await this._execute(
38         type,
39         'update',

```

```

35 async update(type, id, attributes,
36 options = {}) {
37     return await this.authorize(
38         type,
39         ACTIONS.UPDATE,

```


<pre> 1 { type, id, attributes, options 2 }, 3 repository => repository.update(type, id, attributes, options) 4); 5 } </pre>	<pre> repository => repository.update(type, id, attributes, options)) } </pre>
---	--

Response: floragunn denies that it has engaged in any copyright infringement and therefore denies Elastic's allegations in paragraph 47 of the complaint. floragunn further denies that the changes Elastic proposes in paragraph 47 render anything "clear."

Paragraph 48

Allegation: Elastic registered Kibana X-Pack version 6.4.0 with the United States Copyright Office on September 21, 2019.

Response: floragunn admits that Elastic made such a registration, but denies that all its contents are entitled to copyright protection.

Paragraph 49

Allegation: Elastic also previously alleged multiple additional instances of infringement of its copyrights in X-Pack and the Kibana X-Pack plugin in its original complaint and FAC in *floragunn I*.

Response: floragunn admits that Elastic made such allegations, but denies that the allegations have any merit.

Paragraph 50:

Allegation: As alleged in *floragunn I*, infringement by floragunn is also evident in a February 13, 2016 commit to the file SearchGuardFilter.java. That commit and file contains commented out code in at least lines 48–52 that is copied from or is, at least, a derivative work of Elastic code in the file Privilege.java that is original to Elasticsearch Shield versions 1.0.0 Beta1, 1.1.1, and 2.0.0-beta1 and was released in or before 2015.

1 Elastic registered Elasticsearch Shield version 1.0.0 Beta1 with the United States
2 Copyright Office on April 30, 2020 and registered versions 1.1.1 and 2.0.0-beta1 on
3 September 10, 2019.

4 Response: floragunn admits that Elastic made such copyright registrations. floragunn
5 denies that it engaged in copyright infringement or the creation of derivative works, and
6 on that basis denies the all other allegations in paragraph 50.

7 **Paragraph 51:**

8 Allegation: As alleged in *floragunn I*, infringement by floragunn is evident in a May
9 30, 2016 commit to the SearchGuardSSLNettyHttpServerTransport.java file in at least
10 lines 67–88. That code is copied from and/or is a derivative work of Elastic code
11 included in the file ShieldNettyHttpServerTransport.java that was released on June 24,
12 2015 and is original to Elasticsearch Shield version 1.3.0. Elastic registered
Elasticsearch Shield version 1.3.0 with the United States Copyright Office on
September 10, 2019.

13 Response: floragunn denies that it has engaged in any copyright infringement or creation of
14 derivative works and therefore denies Elastic’s allegations in paragraph 51 of the complaint.
15 floragunn further denies that any infringement is “evident.” floragunn specifically denies that
16 any similarities are due to copying of any protectable expression original to Elastic. floragunn
17 admits that Elastic appears to have registered Elasticsearch Shield version 1.3.0 with the United
18 States Copyright Office on September 10, 2019.

21 **Paragraph 52:**

22 Allegation: As alleged in *floragunn I*, infringement by floragunn is evident in a June
23 6, 2016 commit to the Search Guard file FlsFilterLeafReader.java in the search-guard-
24 module-dlsfls repository in at least lines 165–177. That code is copied from and/or is a
25 derivative work of Elastic code in the file FieldSubsetReader.java that was released by
26 Elastic on September 17, 2015 and is original to Elasticsearch Shield version 2.0.0-
27 beta2. Elastic registered Elasticsearch Shield version 2.0.0-beta2 with the United States
28 Copyright Office on September 11, 2019.

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works and therefore denies Elastic’s allegations in paragraph 52 of the complaint. floragunn further denies that any infringement is “evident.” floragunn specifically denies that any similarities are due to copying of any protectable expression original to Elastic. floragunn admits that Elastic appears to have registered Elasticsearch Shield version 2.0.0-beta2 with the United States Copyright Office on September 11, 2019

Paragraph 53:

Allegation: As alleged in *floragunn I*, infringement by floragunn is evident in a March 31, 2018 commit to the Search Guard `get_next_url.js` file in least lines 21–42. That code is copied from and/or is a derivative work of code that Elastic included in a bug fix to the Kibana X-Pack plugin in `parse_next.js` file and that was released on or before April 20, 2017 and is original to Kibana X-Pack version 5.3.1 and Kibana Shield version 2.3.2. Elastic registered Kibana X-Pack version 5.3.1 and Kibana Shield version 2.3.2 with the United States Copyright Office on September 19, 2019 and November 6, 2019, respectively.

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works and therefore denies Elastic’s allegations in paragraph 53 of the complaint. floragunn further denies that any infringement is “evident.” floragunn specifically denies that any similarities are due to copying of any protectable expression original to Elastic. floragunn admits that Elastic appears to have registered Kibana X-Pack version 5.3.1 and Kibana Shield version 2.3.2 with the United States Copyright Office on September 19, 2019 and November 6, 2019, respectively

Paragraph 54:

Allegation: As alleged in *floragunn I*, infringement by floragunn is evident in a June 7, 2018 commit to the file `DlsFlsFilterLeafReader.java`, where floragunn copied the

implementations of at least two methods, `getLiveDocs` and `numDocs`, from X-Pack. floragunn's implementation of `getLiveDocs` and `numDocs` in the June 7, 2018 commit to `DlsFlsFilterLeafReader.java`, found at least in lines 403–473 of that file, is copied from and/or is a derivative work of Elastic's implementation of `getLiveDocs` and `numDocs` in the file `DocumentSubsetReader.java` that Elastic made source-available for the first time in late April 2018 as part of Elasticsearch X-Pack version 6.2.x. This infringed Elastic code is original to Elasticsearch Shield version 2.0.0 RC1, which was released on October 7, 2015 and which Elastic registered with the United States Copyright Office on April 30, 2020.

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works and therefore denies Elastic's allegations in paragraph 54 of the complaint. floragunn further denies that any infringement is "evident." floragunn specifically denies that any similarities are due to copying of any protectable expression original to Elastic..

Paragraph 55:

Allegation: As alleged in *floragunn I*, infringement by floragunn is evident in an October 28, 2018 commit to the Search Guard `get_next_url.js` file in at least lines 22–44. That code is copied from and/or is a derivative work of code that Elastic included in a bug fix to the Kibana X-Pack plugin in `parse_next.js` and that was released on or before January 30, 2018 and is original to Kibana X-Pack versions 5.6.7 and 5.3.1 and Kibana Shield version 2.3.2. Elastic registered Kibana X-Pack versions 5.6.7 and 5.3.1 and Kibana Shield version 2.3.2 with the United States Copyright Office on September 21, 2019, September 19, 2019, and November 6, 2019, respectively.

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works and therefore denies Elastic's allegations in paragraph 55 of the complaint. floragunn further denies that any infringement is "evident." floragunn specifically denies that any similarities are due to copying of any protectable expression original to Elastic. Floragunn admits that Elastic appears to have registered Kibana X-Pack versions 5.6.7 and 5.3.1 and Kibana Shield version 2.3.2 with the United States

Copyright Office on September 21, 2019, September 19, 2019, and November 6, 2019, respectively.

Paragraph 56:

Allegation: As alleged in *floragunn I*, infringement by floragunn is evident in an August 30, 2019 commit to the Search Guard file `call_with_request_factory.js` in at least lines 1–13. That code is copied from and/or is a derivative work of Elastic code that occurs multiple places within the Kibana X-Pack plugin, including in a February 28, 2019 commit to `call_with_request_factory.js`, and that is original to Kibana X-Pack versions 5.4.0 and 7.2.0. Elastic registered Kibana X-Pack versions 5.4.0 and 7.2.0 with the United States Copyright Office on November 5, 2019.

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works and therefore denies Elastic’s allegations in paragraph 56 of the complaint. floragunn further denies that any infringement is “evident.” floragunn specifically denies that any similarities are due to copying of any protectable expression original to Elastic. floragunn admits that Elastic appears to have registered Kibana X-Pack versions 5.4.0 and 7.2.0 with the United States Copyright Office on November 5, 2019.

Paragraph 57:

Allegation: As alleged in *floragunn I*, infringement by floragunn is evident in an August 30, 2019 commit to the Search Guard `fetch_all_from_scroll.js` file in at least lines 1–19. That code is copied from and/or is a derivative work of Elastic code included in `fetch_all_from_scroll.js` that was released on May 4, 2017 and is original to Kibana X-Pack version 5.4.0. Elastic registered Kibana X-Pack version 5.4.0 with the United States Copyright Office on November 5, 2019.

Response: floragunn denies that it has engaged in any copyright infringement or creation of derivative works and therefore denies Elastic’s allegations in paragraph 57 of the complaint. floragunn further denies that any infringement is “evident.” floragunn specifically denies that any similarities are due to copying of any protectable expression

original to Elastic. floragunn admits that Elastic appears to have registered Kibana X-Pack version 5.4.0 with the United States Copyright Office on November 5,.

Paragraph 58:

Allegation: floragunn's marketing and distribution of infringing Search Guard software causes third party Search Guard customers and users to incorporate code that infringes Elastic's copyrights in X-Pack and the Kibana X-Pack plugin. Those third parties therefore necessarily reproduce and use Elastic's proprietary X-Pack and/or Kibana X-Pack plugin code when they incorporate Search Guard into their adoptions of Elasticsearch, thereby infringing Elastic's copyrights.

Response: floragunn denies that Search Guard infringes any copyright and therefore denies Elastic's allegations in paragraph 58 of the complaint.

Paragraph 59:

Allegation: Additional third parties have incorporated floragunn's infringing code into products and services they offer publicly. Elastic has investigated to identify third parties who have incorporated floragunn's infringing code into their products and services.

Response: floragunn denies that Search Guard infringes any copyright and therefore denies Elastic's allegations in paragraph 59 of the complaint.

Paragraph 60:

Allegation: Infringing third party products and services include at least [Amazon.com](https://www.amazon.com), Inc.'s and Amazon Web Services Inc.'s Open Distro for Elasticsearch ("Open Distro") and Amazon Elasticsearch Service ("AESS"), which both contain and/or contained infringing code that originated with floragunn. Open Distro contains and/or contained infringing code that arises from floragunn's infringement of Elastic's copyrights in X-Pack and the Kibana X-Pack plugin. AESS contains or contained infringing code that arises from floragunn's infringement of Elastic's copyrights in X-Pack. Rackspace US, Inc.'s ObjectRocket for Elasticsearch contains or contained infringing code that arises from floragunn's infringement of Elastic's copyrights in X-Pack and the Kibana X-Pack plugin. And IBM Corporation's IBM Cloud Databases for Elasticsearch contains or contained infringing code that arises from floragunn's infringement of Elastic's copyrights in X-Pack.

Response: floragunn denies that Search Guard infringes any copyright and therefore denies Elastic's allegations in paragraph 60 of the complaint.

Paragraph 61:

Allegation: Elastic incorporates by reference each of the allegations in the preceding paragraphs of this Complaint as if fully set forth here.

Response: floragunn repeats and realleges its responses to the complaint as set forth above.

Paragraph 62:

Allegation: As alleged above, on August 14, 2019, Elastic registered with the United States Copyright Office versions 1.0.0 and 2.0.0 of Elasticsearch Shield and versions 5.0.0, 6.0.0, 6.2.0, 6.2.x, and 6.3.0 of X-Pack under Registration Numbers TX 8-762-996, TX 8-762-994, TX 8-762975, TX 8-762-985, TX 8-762-987, TX 8-762-988, and TX 8-762-991, respectively. On September 10, 2019, Elastic registered versions 1.1.1, 1.3.0, 2.0.0-beta1, and, on September 11, 2019, 2.0.0-beta2 of Elasticsearch Shield under Registration Numbers TX 8-773-254, TX 8-773258, TX 8-773-261, and TX 8-773-263, respectively. Elastic additionally registered: version 2.3.2 of the Kibana Shield plugin on November 6, 2019 under Registration Number TX 8-796-945; versions 5.2.0 and 5.3.1 of the Kibana X-Pack plugin on September 19, 2019 under Registration Numbers TX 8-777-406 and TX 8-777-412, respectively; and versions 5.6.7 and 6.4.0 of the Kibana X-Pack plugin on September 21, 2019 under Registration Numbers TX 8-778-023 and TX 8-778-024, respectively; version 5.4.0 of the Kibana X-Pack plugin on November 5, 2019 under Registration Number TX 8-796-010; and version 7.2.0 of the Kibana X-Pack plugin under Registration Number TX 8-796-013 on November 5, 2019. Elastic also registered, on April 30, 2020, Elasticsearch Shield versions 1.0.0 Beta1 and 2.0.0 RC1 and Kibana X-Pack version 5.0.0 under Registration Numbers TXu 2-191-552, TX 8-865-693, and TX 8-865-685, respectively. On October 13, 2020, Elastic further registered version 2.0.1 of Elasticsearch Shield and versions 5.1.1, 5.6.0, and 6.1.1 of Elasticsearch X-Pack under Registration Numbers TX 8-902-217, TX 8902-221, TX 8-902-228, and TX 8-902-227, respectively. Copies of those Certificates of Registration are attached as Exhibits A through Y to this Complaint.

Response: floragunn admits that these registrations appear to have been submitted to the United States Copyright Office.

Paragraph 63:

Allegation: These works contain copyrightable subject matter for which copyright protection exists under the Copyright Act, 17 U.S.C. § 101, *et seq.* elasticsearch B.V. is the exclusive owner of all rights in these copyrighted works. Elasticsearch, Inc. holds the

exclusive license from elasticsearch B.V. to enforce the copyright in and distribute copies of these works in, among other territories, the United States.

Response: floragunn denies that these works contain copyrightable subject matter.

floragunn lacks sufficient information to admit or deny the remaining allegations of paragraph 63.

Paragraph 64

Allegation: Through the actions described herein and in the *floragunn I* complaint and FAC, floragunn has infringed and will continue to infringe Elastic's copyrights in the X-Pack and Kibana X-Pack plugin code by, at least, reproducing (including through creation of intermediate copies), preparing derivative works from (including through decompilation), and distributing copies of those copyrighted works. No license permitted floragunn's infringing activities.

Response: floragunn denies the allegations in paragraph 64.

Paragraph 65

Allegation: floragunn's infringing conduct alleged herein was and continues to be willful and with full knowledge of Elastic's rights in the copyrighted works, and that conduct has enabled floragunn to profit illegally from infringement.

Response: floragunn denies the allegations in paragraph.

Paragraph 66

Allegation: Elastic is entitled to an injunction restraining floragunn, its officers, agents, employees, assigns, and all persons acting in concert with them from engaging in further infringement of Elastic's copyrights.

Response: floragunn denies the allegations in paragraph 66.

Paragraph 67

Allegation: Elastic is entitled to recover from floragunn the damages it has sustained and will sustain as a result of floragunn's wrongful acts as alleged herein. Elastic is further entitled to recover from floragunn the gains, profits, and advantages it has obtained as a result of floragunn's wrongful acts. The full extent of Elastic's damages and the gains, profits, and advantages floragunn has obtained by reason of its aforesaid acts of copyright infringement cannot be determined at this time, but will be proven at trial. Further, Elastic is entitled to recover costs and reasonable attorneys' fees from floragunn as a result of the wrongful acts alleged herein.

Response: floragunn denies the allegations in paragraph 67.

Paragraph 68

Allegation: Elastic incorporates by reference each of the allegations in the preceding paragraphs of this Complaint as if fully set forth here.

Response: floragunn repeats and realleges its responses as set forth above.

Paragraph 69

Allegation: floragunn's distribution of infringing Search Guard software induces, causes, encourages, and materially contributes to Search Guard users and third parties that incorporate Search Guard code into their products and services infringing Elastic's copyrights in the X-Pack and/or Kibana X-Pack plugin code by engaging in unauthorized reproduction and distribution of works containing Elastic's copyrighted material.

Response: floragunn denies the allegations in paragraph 69.

Paragraph 70

Allegation: Elastic is informed and believes, and, on that basis, alleges that floragunn derived substantial financial benefit from Search Guard users' and third parties' infringement of Elastic's copyrights in X-Pack and/or the Kibana X-Pack plugin.

Response: floragunn denies the allegations in paragraph 70.

Paragraph 71

Allegation: floragunn's marketing, commercial distribution of, licensing of, and profit from infringing Search Guard software shows that it knowingly, intentionally, willfully, and purposefully induced, caused, encouraged, and materially contributed to, and continues to knowingly, intentionally, willfully, and purposefully induce, cause, encourage, and materially contributes to, Search Guard users' and third parties' infringement of Elastic's copyrights in X-Pack and/or the Kibana X-Pack plugin.

Response: floragunn denies the allegations in paragraph 71 .

Paragraph 72

Allegation: floragunn has the ability to prevent Search Guard users and third parties from infringing Elastic's copyrights in the X-Pack and Kibana X-Pack plugin code by omitting the infringing code from its Search Guard software product. However, floragunn has not prevented Search Guard users and third parties from infringing Elastic's copyrights in the X-Pack and Kibana X-Pack plugin code.

Response: floragunn denies the allegations in paragraph 72.

Paragraph 73

Allegation: floragunn, through its knowing and intentional inducement, causation, encouragement, and material contribution to the infringement of Elastic's copyrights in the X-Pack and Kibana X-Pack plugin code by Search Guard users and third parties, is committing and/or is contributorily and vicariously liable for the acts of infringement by Search Guard users and third parties. Each act of infringement that floragunn knowingly and intentionally induced, caused, encouraged, and materially contributed to is a separate and distinct act of infringement.

Response: floragunn denies the allegations in paragraph 73.

Paragraph 74

Allegation: Elastic is entitled to an injunction restraining floragunn, its officers, agents, employees, assigns, and all persons acting in concert with them from actions inducing, causing, encouraging, or materially contributing to Search Guard users' and third parties' infringement of Elastic's copyrights.

Response: floragunn denies the allegations in paragraph 74.

Paragraph 75

Allegation: Elastic is entitled to recover from floragunn the damages it has sustained and will sustain as a result of floragunn's acts inducing, causing, encouraging, or materially contributing to Search Guard users' and third parties' infringement of Elastic's copyrights. Elastic is further entitled to recover from floragunn the gains, profits, and advantages it has obtained as a result of its acts inducing, causing, encouraging, or materially contributing to Search Guard users' and third parties' infringement of Elastic's copyrights. The full extent of Elastic's damages and the gains, profits, and advantages floragunn has obtained by reason of its aforesaid acts of copyright infringement by Search Guard users and third parties cannot be determined at this time but will be proven at trial. Further, Elastic is entitled to recover costs and reasonable attorneys' fees from floragunn as a result of the acts inducing, causing, encouraging, or materially contributing to Search Guard users' and third parties' infringement of Elastic's copyrights alleged herein.

Response: floragunn denies the allegations in paragraph 75.

DEFENSES

floragunn denies that any similarities in source code are due to copying of any protectable expression original to Elastic, for at least the reasons stated below. floragunn alleges and asserts the following defenses, undertaking the burden of proof only to the extent that they are deemed affirmative defenses by law. floragunn specifically reserves all rights to allege additional defenses and counterclaims that become known through its investigation into Elastic's allegations in the course of discovery.

First Defense – Independent Creation

1. The Search Guard security plugin was created independently.
2. floragunn's security plugin Search Guard traces its roots to October 2013, when Hendrik Saly, then an independent programmer, but now floragunn's Chief Technology Officer, developed the first complete security plugin solution for the Elasticsearch search engine, appropriately called "Elasticsearch Security Plug-in" ("ESP"). ESP was and is open code and has been publicly visible for all since 2013.

1 3. At the time, Elastic had not developed and was not offering a security plugin for
2 its Elasticsearch search engine.

3 4. In January 2015, Mr. Saly began work on “Elastic Defender” (“Defender”), an
4 advanced security plugin based on ESP.

5 5. In January 2015, Elastic finally released its own security plugin for Elasticsearch
6 called “Shield.” The source code for Shield was not open to the public, and Shield lacked many
7 of the features that had been included in Defender, such as Kerberos, Field Level Security,
8 Document Level Security, Index based output for audit events, Native Realm for storing users,
9 and PKI authentication. In short, Shield was an objectively inferior security product when
10 compared to ESP and its successor Defender.

11 6. That same month, January 2015, Shay Banon, the founder of Elastic, emailed
12 Hendrik Saly, writing that he:
13

14 *Just came across your Elasticsearch security plugin and we are looking for security and*
15 *generally talented engineers with elastic knowledge to joining our company. Interested?*
16 *Up for a quick chat?*
17

18 A period of interviewing ensued, as well as discussions concerning the possible sale of Mr. Saly’s
19 Defender software to Elastic. As part of those discussions, in April 2015, Mr. Saly provided the
20 source code for Defender to Mr. Banon and therefore to Elastic. Defender’s source code had not
21 been made publicly available at that point. Ultimately, Elastic did not offer Mr. Saly employment.

22 7. In May 2015, floragunn acquired an exclusive license from Mr. Saly for the
23 Defender security plugin and set out to improve the code for the product before formally launching
24 Defender rebranded as “Search Guard.” floragunn made the source code for Search Guard
25 available to the public on May 24, 2015.
26

27 8. In June 2015, two months after Mr. Saly provided the source code for Defender to
28 Elastic, and a month after floragunn made the source code for Search Guard public, Elastic released

Shield version 1.3.0, which for the first time contained a PKI authentication feature and an Index Output for audit event feature. These features had never been part of Shield before, but both had previously been included in Defender (the predecessor to floragunn's Search Guard). In October 2015, Elastic released Shield version 2.0, which for the first time, contained the features Field Level Security and Document Level Security features. These features had not been part of Shield in any previous release, but had previously been included in Defender.

9. There can be no doubt that Hendrik Saly and floragunn were the security plugin innovators, and Elastic the follower. Considering floragunn's expertise, and Elastic's late start in the security plugins, there was never any need for floragunn to copy Elastic's code since its security plugin development capabilities already exceeded Elastic's. Moreover, the idea that floragunn would copy garden variety code from Elastic as alleged in its complaint, and then publicly post that code for all to see, and therefore jeopardize its entire business and reputation defies logic. All of floragunn's code was independently created, and therefore does not infringe Elastic's code.

Second Defense – No Copyright Infringement

10. floragunn does not infringe, has not infringed (directly, contributorily, or by inducement), and is therefore not liable for infringement of any valid copyright or copyrights of Elastic, including, without limitations, any copyright rights in the works that are the subject of Elastic's complaint.

11. Among other things, the Elasticsearch search engine and Kibana for which both the Shield (later X-Pack) and Search Guard plugins are created are based on open source code authored by others, such as Lucene, Netty, AngularJS and Node.js. Elasticsearch and Kibana necessarily rely on a variety of code and syntax that was created by others. This means that substantial portions of the Elasticsearch and Kibana computer code are not original to Elastic.

1 12. Like Elasticsearch and Kibana, aspects of Shield and X-Pack are based on code not
2 original to Elastic, including but not limited to opens source libraries or code such as Lucene,
3 Netty, AngularJS, Lodash, and Node.js, and since such aspects are not Elastic's original
4 expression, they are not entitled to copyright protection.

5 **Third Defense -- Elements Not Protected by Copyright**

6
7 13. Elasticsearch is designed specifically to allow the creation of plugins like Shield
8 and Search Guard to extend its functionality. Elastic has long documented the specific means
9 that developers should follow when creating Elasticsearch plugins. The requirements for a plugin
10 to operate necessarily constrain the choices made by plugin developers. This is analogous to
11 third-party upgrade kits for physical appliances; for example, a turbocharger for a car necessarily
12 will need to physically mate with the engine of the car, and thus the design choices for a
13 turbocharger for a Corvette will necessarily be constrained by the physical design of a Corvette.
14 Search Guard likewise must behave in certain ways in order for it to operate with Elasticsearch.

15
16 14. Elastic's own Shield product likewise is a plugin for Elasticsearch, and thus the
17 design choices for Shield were also constrained by the need to operate with Elasticsearch. Just as
18 two turbochargers designed to work with a Corvette will share certain similarities because both
19 must attach to the same engine, Search Guard and Shield likewise must share some similarities
20 because both are plugins for the same Elasticsearch product.

21
22 15. Furthermore, Search Guard and Shield (X-Pack) both provide similar
23 functionality for certain features (though the Search Guard product offers more functionality than
24 Elastic's security plugin). That similarity in functionality acts as a further constraint on the
25 design choices that floragunn and Elastic respectively made in designing their products.
26 Returning to the turbocharger analogy, any turbocharger necessarily will include a turbine—that
27 is because turbochargers use turbines, not because one turbocharger is a copy of another.
28

1 16. In addition, there are many “tools of the trade” that are known to many in the
2 developer community, and that draw on tropes common to computer programming. These tropes
3 can result in superficial similarities between independently developed code, especially when the
4 code is reviewed by a lay observer who is not familiar with the common programming
5 conventions that are known to software engineers.

6
7 17. Elastic’s copyright claims are barred to the extent that Elastic claims rights to
8 elements of Elastic’s software or other works that are functional, are not original, or are
9 otherwise not protectable by copyright or are otherwise not protected by the registered
10 copyrights identified in the complaint.

11 **Fourth Defense – Fair Use**

12 18. Elastic’s claims for copyright infringement are barred in whole or part by the
13 doctrine of fair use pursuant to 17 U.S.C. § 107 in view of the nature of works asserted by Elastic
14 and covered by the copyrights identified in the complaint, the amount (if any) and substantiality
15 of the portions of such works used by floragunn (if any), in relations to the works as a whole, the
16 purpose and character of any use thereof by floragunn, and the effect, if any, of such use on the
17 potential market for the works.

18 **Fifth Defense – *De Minimis* Copying**

19 19. Elastic’s claims for copyright infringement are barred by the doctrine of de
20 minimis copying, as any alleged copying of protectable portions of the work that are the subject
21 of the claimed copyrights was *de minimis*.

22 **Sixth Defense – No Intent to Induce Copyright Infringement**

23 20. floragunn has not engaged in purposeful, culpable expression or conduct designed
24 or intended to result in others infringing Elastic’s alleged copyrights and thus is not liable under
25 Elastic’s inducement claims.
26
27
28

Seventh Defense -- No Injunctive Relief

21. Elastic has not suffered any irreparable injury, and has an adequate remedy at law, and injunctive relief is unwarranted because it would be contrary to the public interest.

Eighth Defense – Statute of Limitations

22. Elastic's claims for damages are barred in part by the applicable statute of limitations.

DATED: December 15, 2020

WUERSCH & GERING LLP

By: /s/ V. David Rivkin /s/
V. DAVID RIVKIN (admitted *pro hac vice*)
david.rivkin@wg-law.com
JOHN A. SMITTEN (admitted *pro hac vice*)
john.smitten@wg-law.com
100 Wall St., 10th Fl.
New York, NY 10005
Telephone: 212 509-5050
Facsimile: 212 509-9559

KWUN BHANSALI LAZARUS LLP
MICHAEL S. KWUN (SBN 198945)
mkwun@kblfirm.com
555 Montgomery St., Suite 750
San Francisco, CA 94111
Telephone: 415 630-2350
Facsimile: 415 367-1539

Attorneys for Defendant
FLORAGUNN GmbH

DEMAND FOR JURY TRIAL

In accordance with Rule 38 of the Federal Rules of Civil Procedure and Civil L.R. 3-6(a),
floragunn GmbH respectfully demands a jury trial of all issues triable to a jury in this action.

DATED: December 15, 2020

WUERSCH & GERING LLP

By: /s/ V. David Rivkin /s/
V. DAVID RIVKIN (admitted pro hac vice)
david.rivkin@wg-law.com
JOHN A. SMITTEN (admitted pro hac vice)
john.smitten@wg-law.com
100 Wall St., 10th Fl.
New York, NY 10005
Telephone: 212 509-5050
Facsimile: 212 509-9559

KWUN BHANSALI LAZARUS LLP
MICHAEL S. KWUN (SBN 198945)
mkwun@kblfirm.com
555 Montgomery St., Suite 750
San Francisco, CA 94111
Telephone: 415 630-2350
Facsimile: 415 367-1539

Attorneys for Defendant
FLORAGUNN GmbH

CERTIFICATE OF SERVICE

I am an attorney at Wuersch & Gering LLP, counsel for Defendant, in the above-captioned proceeding. I hereby certify that on December 15, 2020, I caused the foregoing **Answer to Plaintiffs' Complaint** to be served electronically via CM/ECF upon Plaintiffs Elasticsearch, Inc. and elasticsearch, B.V.

/s/ John A. Smitten /s/

John A. Smitten